

Amendments to the Claims

This listing of claims will replace all prior versions, and listings, of claims in the application:

Listing of Claims:

1. (currently amended) An apparatus within a pipelined microprocessor for forwarding store instruction results to a pipeline stage for execution of a load instruction, the apparatus comprising:
 - a result forwarding cache (RFC), for storing at least one non-store instruction result destined for a user-visible register of the microprocessor, and for storing a plurality of store instruction results destined for a data cache of the microprocessor;
 - comparison logic, for comparing a load address of the load instruction with a plurality of store addresses associated with said plurality of store instruction results to generate an address match signal; and
 - control logic, configured to receive said match signal and selectively forward one of said plurality of store instruction results from said RFC to the pipeline stage in response to said match signal, thereby avoiding stalling the load instruction until said one of said plurality of store instruction results is updated into said data cache.
2. (original) The apparatus of claim 1, wherein said plurality of store instruction results comprise data to be stored from the microprocessor into a memory attached thereto.
3. (original) The apparatus of claim 1, wherein said load address specifies a location of data to be loaded into the microprocessor from a memory attached thereto.
4. (original) The apparatus of claim 1, wherein said RFC comprises a plurality of storage elements for storing a predetermined number of instruction results.

5. (original) The apparatus of claim 4, wherein said instruction results are received by said RFC from an execution unit of the microprocessor.
6. (original) The apparatus of claim 4, wherein said plurality of storage elements store said predetermined number of instruction results in a first-in-first-out manner.
7. (original) The apparatus of claim 4, wherein said predetermined number of instruction results is five.
8. (original) The apparatus of claim 1, wherein said load address and said plurality of store addresses comprise virtual addresses.
9. (original) The apparatus of claim 8, wherein said virtual addresses comprise x86 linear addresses.
10. (currently amended) An apparatus for forwarding storehit data within stages of a pipelined microprocessor, the apparatus comprising:
 - a result forwarding cache (RFC), configured to forward at least one non-store instruction result, and to store and forward a first plurality of store instruction results destined for a data cache of the microprocessor;
 - a data unit, configured to forward a second plurality of store instruction results;
 - and
 - selection logic, coupled to said RFC and said data unit, for selectively providing one of said first and second plurality of store instruction results to a stage of the microprocessor pipeline executing a load instruction.
11. (original) The apparatus of claim 10, wherein said load instruction comprises a load address for specifying an address of data to be loaded into the microprocessor, wherein said selection logic is configured to forward one of said first and second plurality of store instruction results only if said load address matches one or more of a first and second plurality of store addresses corresponding to said first and second plurality of store instruction results.

12. (original) The apparatus of claim 11, wherein selection logic forwards said first plurality of store instruction results forwarded by said RFC at a higher priority than said second plurality of store instructions results forwarded by said data unit if said load address matches both one or more of said first plurality of store addresses and one or more of said second plurality of store addresses.
13. (original) The apparatus of claim 11, further comprising:
comparison logic, coupled to said selection logic, for comparing said load address with said first and second plurality of store addresses to determine whether said load address matches one or more of said first and second plurality of store addresses.
14. (original) The apparatus of claim 11, wherein said data unit is configured to forward said second plurality of store instruction results from a plurality of store buffers of the microprocessor.
15. (original) The apparatus of claim 14, wherein said plurality of store buffers is configured to store said second plurality of store instruction results while said second plurality of store instruction results are written to a memory coupled to the microprocessor.
16. (original) The apparatus of claim 14, wherein said data unit is configured to forward a newest one of said second plurality of store instruction results if said load address matches more than one of said second plurality of store addresses.
17. (original) The apparatus of claim 11, wherein said RFC is configured to forward a newest one of said first plurality of store instruction results if said load address matches more than one of said first plurality of store addresses.

18. (previously presented) An apparatus for detecting storehit conditions in a pipelined microprocessor in a hierarchical manner, the apparatus comprising:
- first comparison logic, for comparing a load instruction load address in a first stage of the pipeline with a first plurality of store addresses of first store instruction data in a plurality of stages of the pipeline subsequent to said first pipeline stage, wherein said plurality of stages of the pipeline subsequent to said first pipeline stage comprise non-store buffers;
 - second comparison logic, for comparing said load address with a second plurality of store addresses of second store instruction data in a plurality of store buffers of the microprocessor; and
 - control logic, coupled to said first and second comparison logic, configured to determine which of said first and second store instruction data is newest based on said first and second comparison logic comparing, and to forward said newest store instruction data to said first pipeline stage in response thereto.
19. (original) The apparatus of claim 18, wherein said first comparison logic is configured to compare virtual addresses.
20. (original) The apparatus of claim 18, wherein said second comparison logic is configured to compare physical addresses.

21. (currently amended) An apparatus for speculatively forwarding storehit data in a microprocessor pipeline, the apparatus comprising:

~~a plurality of first and second~~ virtual address comparators, for comparing a virtual load address with ~~a plurality of first and second~~ virtual store addresses to generate a virtual match signal for indicating whether first and second storehit data is likely present in a store buffer and a result forwarding cache, respectively, of the microprocessor, wherein if said first and second storehit data are both present said second storehit data is newer than said first storehit data;

~~a plurality of first and second~~ physical address comparators, for comparing a physical load address translated from said virtual load address with ~~a plurality of first and second~~ physical store addresses translated from said plurality of virtual store addresses to generate a physical match signal for indicating whether said first and second storehit data is certainly present in said store buffer and said result forwarding cache, respectively;

forwarding logic, coupled to receive said virtual match signal, for forwarding ~~the~~ said second storehit data present in said store buffer in response to said virtual match signal indicating no match between said virtual load address and said ~~plurality of second~~ virtual store address[[es]] but a match between said virtual load address and said first virtual store address, prior to generation of said physical match signal; and

control logic, for receiving said virtual and physical match signals and generating a stall signal for stalling the pipeline subsequent to said forwarding logic forwarding said storehit data from said store buffer if said physical match signal indicates a match between said physical load address and ~~one of~~ said ~~plurality of second~~ physical store address[[es]] but although said virtual match signal previously indicated ~~indicates~~ no match between said virtual load address and ~~one of~~ said ~~plurality of second~~ virtual store address[[es]], until correct data specified by said physical load address is provided to replace said previously forwarded second storehit data.

22. (currently amended) The apparatus of claim 21, further comprising:
a data unit, configured to forward said correct data specified by ~~the~~said physical
load address to replace said previously forwarded second storehit data;
wherein said control logic is configured to deassert said stall signal after said data
unit forwards said correct data.
23. (previously presented) A pipelined microprocessor for speculatively forwarding
storehit data from a first pipeline stage to a second pipeline stage, wherein the
storehit data is specified by a load address in the second stage, comprising:
address region logic, configured to receive the load address and generate a match
signal to indicate whether the load address is within one of a plurality of
non-cacheable address regions of the microprocessor address space stored
therein;
forwarding logic, for forwarding the storehit data from the first stage to the
second stage prior to said address region logic generating said match
signal;
control logic, configured to receive said match signal and to assert a stall signal,
subsequent to said forwarding logic forwarding the storehit data, to stall
the pipeline if said address region logic indicates the load address is within
one of said plurality of non-cacheable address regions, and to
subsequently obtain using the load address non-cached correct data from a
device external to the microprocessor, for provision to the second stage.
24. (original) The microprocessor of claim 23, further comprising:
a bus interface unit, for receiving data from a bus coupled to the microprocessor,
said bus further coupled to a system memory and a plurality of peripheral
devices; and
at least one response buffer, operatively coupled to the second stage, for receiving
load data specified by the load address from said bus interface unit, and
for providing said load data to the second stage to replace the storehit data
if the load address is within one of said plurality of non-cacheable address
regions.

25. (original) The microprocessor of claim 23, wherein said plurality of non-cacheable regions stored in said address region logic are software-programmable.
26. (previously presented) A method for forwarding storehit data in a microprocessor pipeline, the method comprising:
storing at least one store instruction result and at least one non-store instruction result into a result forwarding cache of the microprocessor;
detecting a storehit condition, wherein a load instruction in a stage of the pipeline specifies data generated by a previous store instruction, wherein said data is still present in the pipeline;
determining whether said data is present in said result forwarding cache;
selectively forwarding said data from said result forwarding cache to said stage if said data is in said result forwarding cache; and
selectively forwarding said data from a data unit of the microprocessor to said stage if said data is not in said result forwarding cache.
27. (original) The method of claim 26, further comprising:
storing results data of each store instruction executed by an execution unit of the microprocessor in said result forwarding cache.
28. (original) The method of claim 26, wherein said detecting said storehit condition comprises:
comparing an address of said data specified by said load instruction with a plurality of store instruction result data addresses stored in the pipeline below said stage; and
determining said address matches one or more of said plurality of data addresses.
29. (original) The method of claim 26, wherein said determining whether said data is present in said result forwarding cache comprises:
comparing an address of said data specified by said load instruction with a plurality of store instruction result data addresses stored in a predetermined number of stages of the pipeline below said stage;

wherein said predetermined number equals a number of result entries in said result forwarding cache.

30. (currently amended) A method for speculatively forwarding storehit data in a microprocessor pipeline, the method comprising:

determining that a virtual load address matches ~~no~~ a first virtual store address present in the pipeline to indicate first storehit data is likely present in a store buffer of the microprocessor, but does not match a second virtual store address present in the pipeline to indicate second newer storehit data is likely absent in a result forwarding cache of the microprocessor;

forwarding the first storehit data from a first stage comprising the store buffer to a second stage of the pipeline having a load instruction specifying the load address based on said determining that ~~said virtual load address matches no virtual store addresses present in the pipeline~~ the first storehit data is likely present in the store buffer and the second newer storehit data is likely absent in the result forwarding cache;

detecting that a physical load address translated from said virtual load address matches a physical store address translated from the second virtual store address to indicate the second newer storehit data is certainly present in the result forwarding cache ~~present in the pipeline~~, subsequent to said forwarding ~~said the first~~ storehit data; and

stalling the pipeline in response to said detecting that said physical load address translated from said virtual load address matches said physical store address present in the pipeline, until correct data specified by said physical load address is provided to replace the previously forwarded second newer storehit data.

31. (original) The method of claim 30, further comprising:

forwarding correction data from a third stage of the pipeline to said second stage after said stalling the pipeline; and

unstalling the pipeline after said forwarding said correction data.

32. (canceled)

33. (original) The method of claim 30, wherein said storehit data comprises a store instruction result within the pipeline having an identical physical store address as said physical load address.

34. (previously presented) A method for speculatively forwarding storehit data in a microprocessor pipeline, the method comprising:
detecting a storehit condition by comparing a load address with a plurality of store addresses;
forwarding storehit data in response to said detecting said storehit condition;
determining said load address is within a non-cacheable address region subsequent to said speculatively forwarding; and
stalling the pipeline in response to said determining said load address is within a non-cacheable address region.

35. (new) A method for speculatively forwarding storehit data in a microprocessor pipeline, the method comprising:
- comparing a virtual load address with first and second virtual store addresses, wherein a load instruction specifying the virtual load address is newer than a first store instruction specifying the first virtual store address, which is newer than a second store instruction specifying the second virtual store address;
 - speculatively forwarding a result of the first store instruction to the load instruction, in response to said comparing indicating the virtual load address matches the first virtual store address and mismatches the second virtual store address;
 - comparing a physical load address with a physical store address, wherein the physical load address is a translation of the virtual load address, wherein the physical store address is a translation of the second virtual store address;
 - determining said forwarding the result of the first store instruction to the load instruction was incorrect, after said speculatively forwarding the result of the first store instruction, in response to said comparing indicating the physical load address matches the physical store address; and
 - forwarding a result of the second store instruction to the load instruction, in response to said determining.
36. (new) The method of claim 35, wherein said comparing the virtual load address with the first virtual store address comprises determining whether the virtual load address hits in a data cache of the microprocessor, wherein said speculatively forwarding the result of the first store instruction to the load instruction comprises the data cache providing the result of the first store instruction to the load instruction.

37. (new) The method of claim 35, further comprising:
- stalling the pipeline, in response to said determining, until said forwarding the result of the second store instruction to the load instruction.
38. (new) The method of claim 35, further comprising:
- writing the result of the second store instruction to a data cache of the microprocessor, in response to said determining;
- wherein said forwarding the result of the second store instruction to the load instruction comprises the data cache providing the result of the second store instruction to the load instruction.
39. (new) The method of claim 35, further comprising:
- reissuing the load instruction, in response to said determining.
40. (new) The method of claim 39, wherein said reissuing the load instruction comprises providing the virtual load address to a data cache of the microprocessor from a replay buffer of the microprocessor.
41. (new) The method of claim 35, wherein said comparing the virtual load address with the second virtual store address is performed prior to the result of the second store instruction being stored a store buffer of the microprocessor.
42. (new) The method of claim 35, wherein said speculatively forwarding the result of the second store instruction to the load instruction comprises forwarding the result of the second store instruction from a result forwarding cache (RFC) of the microprocessor.
43. (new) The method of claim 42, further comprising:
- caching the result of the second store instruction in the RFC, prior to said comparing the virtual load address with the second virtual store address.

44. (new) The method of claim 43, further comprising:

 caching a result of a non-store instruction in the RFC, prior to said comparing the
 virtual load address with the second virtual store address.